Data Visualization and Webscraping

Matt Prest

March 19, 2019

## R Markdown

#Data Visualization  
library(ggplot2)  
library(dplyr)

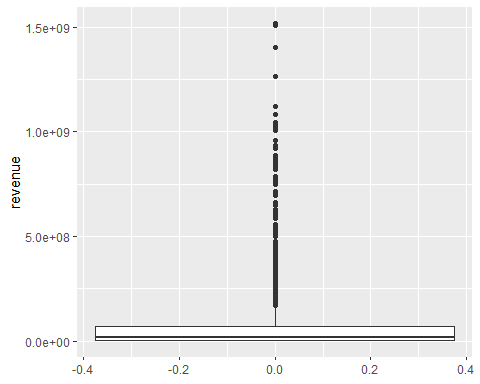
## Warning: package 'dplyr' was built under R version 3.5.2

##   
## Attaching package: 'dplyr'

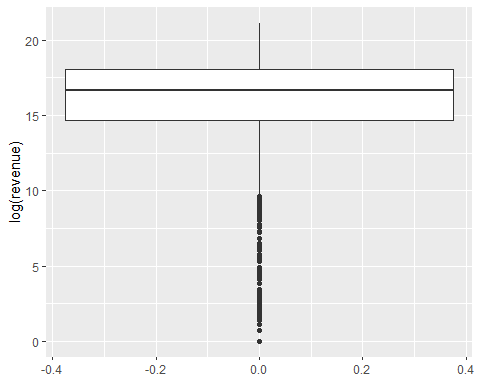
## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

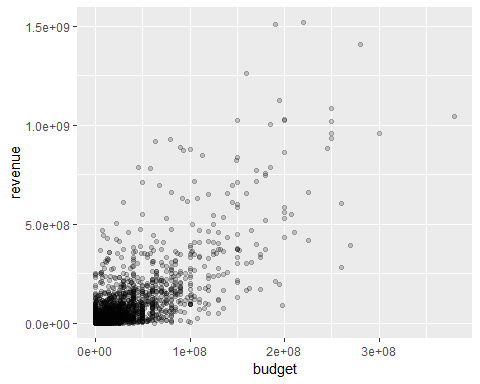
tmbd\_train <- read.csv("C:\\Users\\Matt\\Desktop\\Stats Stuff\\Applied Regression\\TMBD\\train.csv",   
 stringsAsFactors = FALSE)  
tmbd\_test <- read.csv("C:\\Users\\Matt\\Desktop\\Stats Stuff\\Applied Regression\\TMBD\\test.csv",   
 stringsAsFactors = FALSE)  
  
#Revenue boxplot  
ggplot(aes(y = revenue), data = tmbd\_train) + geom\_boxplot()



#clearly heavily skewed  
  
#logarithmic revenue boxplot  
ggplot(aes(y = log(revenue)), data = tmbd\_train) + geom\_boxplot()

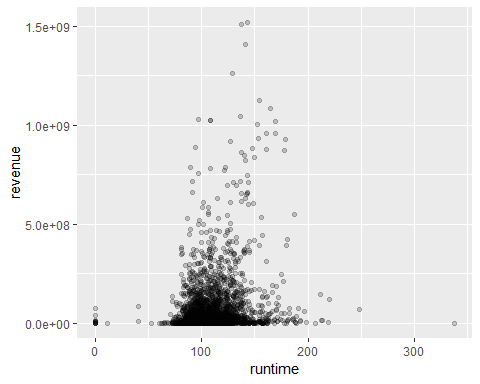


#Revenue vs Budget  
ggplot(aes(x = budget, y = revenue), data = tmbd\_train) + geom\_point(alpha = 0.2)

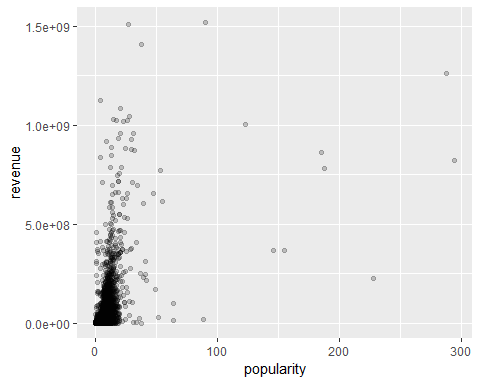


#Revenue vs Runtime  
ggplot(aes(x = runtime, y = revenue), data = tmbd\_train) + geom\_point(alpha = 0.2)

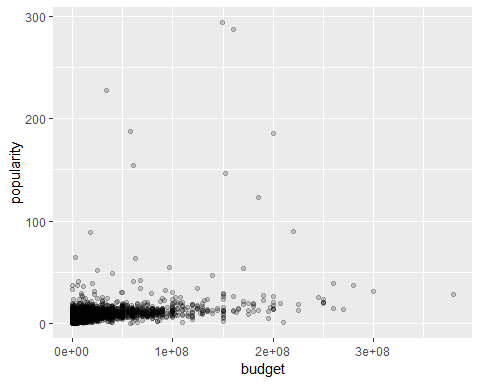
## Warning: Removed 2 rows containing missing values (geom\_point).



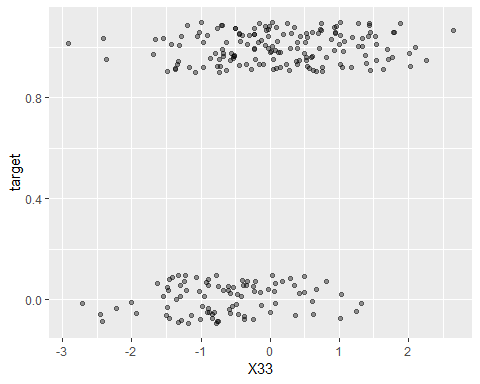
#Revenue vs Popularity  
ggplot(aes(x = popularity, y = revenue), data = tmbd\_train) + geom\_point(alpha = 0.2)



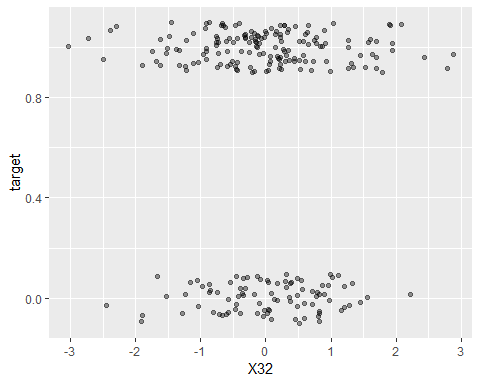
#Popularity vs Budget  
ggplot(aes(x = budget, y = popularity), data = tmbd\_train) + geom\_point(alpha = 0.2)



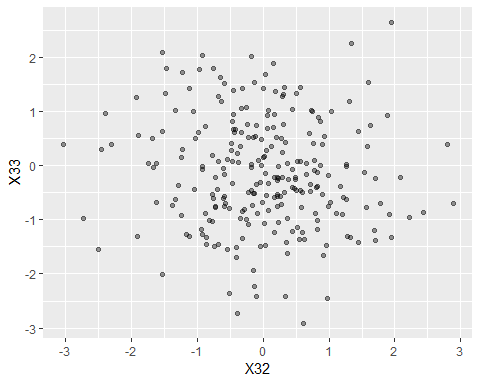
##################################### Don't Overfit ############################################  
dat <- read.csv("C:\\Users\\Matt\\Desktop\\Stats Stuff\\Applied Regression\\train.csv", stringsAsFactors = F)  
test <- read.csv("C:\\Users\\Matt\\Desktop\\Stats Stuff\\Applied Regression\\test.csv", stringsAsFactors = F)  
  
#Target vs highest correlation predictor  
ggplot(aes(x = X33, y = target), data = dat) + geom\_jitter(alpha = 0.4, width = 0, height = 0.1)



#Target vs random low correlation predictor  
ggplot(aes(x = X32, y = target), data = dat) + geom\_jitter(alpha = 0.4, width = 0, height = 0.1)



#Testing independence between arbitrary predictors  
ggplot(aes(x = X32, y = X33), data = dat) + geom\_point(alpha = 0.4)



################################## Webscraping (Do not run!) ################################  
# #web scraping  
# library(rvest)  
# library(tidyverse)  
# library(data.table)  
# library(caret)  
# library(klaR)  
# library(pls)  
# library(dplyr)  
# dd <- read.csv("C:\\Users\\Matt\\Desktop\\Stats Stuff\\Applied Regression\\TMBD\\train.csv",   
# stringsAsFactors = FALSE)  
# tt <- read.csv("C:\\Users\\Matt\\Desktop\\Stats Stuff\\Applied Regression\\TMBD\\test.csv",   
# stringsAsFactors = FALSE)  
#   
#   
# director <- rbindlist(lapply(dd$imdb\_id[1:3000], function(title){  
# url1 <- paste0("https://www.imdb.com/title/", title, "/")  
# web\_page <- read\_html(x = url1)  
# cast <- html\_node(web\_page, css = ".credit\_summary\_item")  
# director <- html\_text(cast) %>%  
# str\_remove\_all("\\n") %>%  
# str\_remove("^\\s+") %>%  
# str\_remove("\\s+$") %>%  
# str\_remove\_all("Directors") %>%  
# str\_remove\_all("Director") %>%  
# str\_remove\_all(":") %>%  
# str\_remove\_all("\\(uncredited\\)") %>%  
# str\_split\_fixed(", ", n = 2)  
#   
#   
# data.frame(imdb\_id = title, director)  
# }))  
#   
# director2 <- rbindlist(lapply(tt$imdb\_id[1:4398], function(title){  
# url1 <- paste0("https://www.imdb.com/title/", title, "/")  
# web\_page <- read\_html(x = url1)  
# cast <- html\_node(web\_page, css = ".credit\_summary\_item")  
# director <- html\_text(cast) %>%  
# str\_remove\_all("\\n") %>%  
# str\_remove("^\\s+") %>%  
# str\_remove("\\s+$") %>%  
# str\_remove\_all("Directors") %>%  
# str\_remove\_all("Director") %>%  
# str\_remove\_all(":") %>%  
# str\_remove\_all("\\(uncredited\\)") %>%  
# str\_split\_fixed(", ", n = 2)  
#   
#   
# data.frame(imdb\_id = title, director)  
# }))  
#   
# #director processing  
# summary(director2)  
# mutate(director2, X1 = ifelse(exists(director2$X1, where = director$X1, inherits = F),   
# X1, factor("other")))  
#   
# dd <- dd %>% dplyr::select(revenue, budget, popularity) %>%  
# mutate(budget = ifelse(budget <= 1000, budget + 1000, budget)) %>%  
# mutate(budget = log(budget)) %>%  
# mutate(revenue = log(revenue))  
#   
# tt <- tt %>%  
# mutate(budget = ifelse(budget == 0, 1000, budget)) %>%  
# mutate(budget = log(budget))  
#   
# dd <- dd %>%  
# data.frame(Director = director$X1)  
# tt <- tt %>%  
# data.frame(Director = director2$X1)